

## **CLAIM AMENDMENTS**

### **Claim Amendment Summary**

#### **Claims pending**

- Before this Amendment: Claims 1-53.
- After this Amendment: Claims 1-53.

**Non-Elected, Canceled, or Withdrawn claims:** None.

**Amended claims:** 1-27 and 33-53.

**New claims:** None.

---

### **Claims:**

**1. (Currently Amended)** One or more processor-accessible tangible storage media comprising processor-executable instructions stored thereon that, when executed, direct a device to perform a method comprising:

provide providing a programming interface for developing programs, the programming interface having multiple groups of types, the programming interface comprising and including:

a first group of types related to core file system concepts;

a second group of types related to entities that a human being can contact;

a third group of types related to documents;

a fourth group of types common to multiple kinds of media;  
a fifth group of types specific to audio media;  
a sixth group of types specific to video media;  
a seventh group of types specific to image media;  
an eighth group of types specific to electronic mail messages; and  
a ninth group of types related to identifying particular locations,  
wherein the programming interface provides callable multiple functions,  
wherein each one of the multiple groups of types provides a corresponding  
set of related ones of the multiple functions.

**2. (Currently Amended)** A ~~programming interface tangible storage~~  
~~media~~ as recited in claim 1, ~~wherein the programming interface~~ further  
compris[[ing]]es: a tenth group of types related to moving data between file  
systems.

**3. (Currently Amended)** A ~~programming interface tangible storage~~  
~~media~~ as recited in claim 1, ~~wherein the programming interface~~ further  
compris[[ing]]es: a tenth group of types related to creating and managing rules  
for generating notifications.

**4. (Currently Amended)** A programming interface tangible storage media as recited in claim 1, wherein the programming interface further compris[[ing]]es: a tenth group of types describing types defined in all the other groups of types.

**5. (Currently Amended)** A programming interface tangible storage media as recited in claim 1, wherein the programming interface further compris[[ing]]es: a tenth group of types related to base types that form a foundation to support all the other groups of types.

**6. (Currently Amended)** A programming interface tangible storage media as recited in claim 1, wherein the programming interface further compris[[ing]]es: a tenth group of types common to multiple kinds of messages, including the electronic mail messages; and an eleventh group of types specific to facsimile messages.

**7. (Currently Amended)** A programming interface tangible storage media as recited in claim 1, wherein the programming interface further compris[[ing]]es: a tenth group of types related to annotations; and an eleventh group of types related to notes;

**8. (Currently Amended)** A ~~programming interface~~ tangible storage media as recited in claim 1, ~~wherein the programming interface~~ further compris[[ing]]es: a tenth group of types related to installed programs; and an eleventh group of types related to installed games.

**9. (Currently Amended)** A ~~programming interface~~ tangible storage media as recited in claim 1, ~~wherein the programming interface~~ further compris[[ing]]es: a tenth group of types related to actions taken by a user; and an eleventh group of types related to maintaining and accessing help information.

**10. (Currently Amended)** A ~~programming interface~~ tangible storage media as recited in claim 1, ~~wherein the programming interface~~ further compris[[ing]]es: a tenth group of types related to a natural language search engine.

**11. (Currently Amended)** A ~~programming interface~~ tangible storage media as recited in claim 1, ~~wherein the programming interface~~ further compris[[ing]]es: a tenth group of types related to tasks in a user interface to let a user know what actions the user can perform when navigating the user interface.

**12. (Currently Amended)** A programming interface tangible storage media as recited in claim 1, wherein the programming interface further compris[[ing]]es: a tenth group of types related to user tasks.

**13. (Currently Amended)** A programming interface tangible storage media as recited in claim 1, wherein the programming interface further compris[[ing]]es: a tenth group of types related to services that can be accessed.

**14. (Currently Amended)** A programming interface tangible storage media as recited in claim 13, wherein the services can be accessed over a network.

**15. (Currently Amended)** A programming interface tangible storage media as recited in claim 1, wherein the programming interface further compris[[ing]]es: a tenth group of types related to identifying access rights.

**16. (Currently Amended)** A programming interface tangible storage media as recited in claim 1, wherein the programming interface further compris[[ing]]es: a tenth group of types related to calendar types.

**17. (Currently Amended)** A programming interface tangible storage media as recited in claim 1, wherein the programming interface further compris[[ing]]es: a tenth group of types related to creating and managing event monitoring and resultant actions.

**18. (Currently Amended)** A programming interface tangible storage media as recited in claim 1, wherein the programming interface further compris[[ing]]es: a tenth group of types used for interop for each of the first through ninth groups of types.

**19. (Currently Amended)** A programming interface tangible storage media as recited in claim 1, wherein the programming interface further compris[[ing]]es: an additional group of types for each of the first through ninth groups of bytes, wherein each of the additional groups of types are for interop.

**20. (Currently Amended)** A programming interface tangible storage media as recited in claim 1, wherein the programming interface further compris[[ing]]es: a tenth group of types related to files stored in a file system.

**21. (Currently Amended)** A programming interface tangible storage media as recited in claim 1, wherein the programming interface further compris[[ing]]es: a tenth group of types related to a category hierarchy.

**22. (Currently Amended)** A system comprising:  
means for exposing via a programming interface for developing programs a first set of functions that represent core concepts of a file system of the system;

means for exposing via the programming interface a second set of functions that enable maintaining information regarding entities that can be contacted; and

means for exposing via the programming interface a third set of functions that allow document types to be accessed.

**23. (Currently Amended)** A system as recited in claim 22, further comprising means for exposing via the programming interface a fourth set of functions related to base types for a plurality of kinds of media; means for exposing via the programming interface a fifth set of functions related specifically to audio media; and means for exposing via the programming interface a sixth set of functions related specifically to video media.

**24. (Currently Amended)** A system as recited in claim 23, further comprising: means for exposing via the programming interface a seventh set of functions related specifically to image media.

**25. (Currently Amended)** A system as recited in claim 22, further comprising means for exposing via the programming interface a fourth set of functions related specifically to electronic mail messages.

**26. (Currently Amended)** A system as recited in claim 22, further comprising means for exposing via the programming interface a fourth set of functions that enable maintaining physical location information.

**27. (Currently Amended)** A method of organizing a set of types for a file system in a program development computer system into a hierarchical namespace, the file system being one of multiple groups of types included in a programming interface for developing programs, the method comprising:

creating a plurality of groups from the set of types using the program development computer system, each group containing at least one type that exposes logically related functionality of the programming interface;

assigning a name to each group in the plurality using the program development computer system, wherein one of the groups in the plurality



includes functionality related to core concepts of the file system, wherein another of the groups in the plurality includes functionality related to entities that a human being can contact, wherein another of the groups in the plurality includes functionality related to document types that can be stored in the file system, and wherein another of the groups in the plurality includes functionality related to multiple kinds of media; and

selecting a top level identifier and prefixing the name of each group with the top level identifier using the program development computer system so that the types in each group are referenced by a hierarchical name that includes the selected top level identifier prefixed to the name of the group containing the type.

**28. (Original)** A method as recited in claim 27, wherein another of the groups in the plurality includes functionality particularly for audio media, wherein another of the groups in the plurality includes functionality particularly for video media, and wherein another of the groups in the plurality includes functionality particularly for image media.

**29. (Original)** A method as recited in claim 27, wherein another of the groups in the plurality includes functionality related to electronic mail.

**30. (Original)** A method as recited in claim 27, wherein another of the groups in the plurality includes functionality related to maintaining physical location information.

**31. (Original)** A method as recited in claim 27, wherein the assigning comprises: assigning a name of Core to the group that includes functionality related to core concepts of the file system so that the hierarchical name for the group that includes functionality related to core concepts of the file system is System.Storage.Core; assigning a name of Contacts to the group that includes functionality related to entities that a human being can contact so that the hierarchical name for the group that includes functionality related to entities that a human being can contact is System.Storage.Contacts; assigning a name of Documents to the group that includes functionality related to document types that can be stored in the file system so that the hierarchical name for the group that includes functionality related to document types that can be stored in the file system is System.Storage.Documents; and assigning a name of Media to the group that includes functionality related to multiple kinds of media so that the hierarchical name for the group that includes functionality related to multiple kinds of media is System.Storage.Media.

**32. (Original)** A method as recited in claim 27, wherein the assigning comprises: assigning a name of Core to the group that includes functionality related to core concepts of the file system so that the hierarchical name for the group that includes functionality related to core concepts of the file system is System.Storage.Core; assigning a name of Contact to the group that includes functionality related to entities that a human being can contact so that the hierarchical name for the group that includes functionality related to entities that a human being can contact is System.Storage.Contact; assigning a name of Document to the group that includes functionality related to document types that can be stored in the file system so that the hierarchical name for the group that includes functionality related to document types that can be stored in the file system is System.Storage.Document; and assigning a name of Media to the group that includes functionality related to multiple kinds of media so that the hierarchical name for the group that includes functionality related to multiple kinds of media is System.Storage.Media.

**33. (Currently Amended)** A method for organizing a file system in a program development computer system, the method comprising:

creating a first namespace with functions that enable identification of particular physical locations using the program development computer system;

and

creating a second namespace with functions that enable identification of entities that can be contacted by a human being using the program development computer system, wherein the first namespace and the second namespace are included in the file system, the file system being included in a programming interface.

**34. (Currently Amended)** A method as recited in claim 33, further comprising: creating a third namespace using the program development computer system with functions that enable documents to be described.

**35. (Currently Amended)** A method as recited in claim 33, further comprising: creating a third namespace using the program development computer system with functions specific to electronic mail messages.

**36. (Currently Amended)** A method as recited in claim 33, further comprising: creating a third namespace using the program development computer system with functions common to multiple kinds of media; creating a fourth namespace using the program development computer system with functions specific to audio media; creating a fifth namespace using the program development computer system with functions specific to video media; and

creating a sixth namespace using the program development computer system with functions specific to image media.

**37. (Currently Amended)** A method as recited in claim 33, further comprising: creating a third namespace using the program development computer system with functions that are expected to be used by all other namespaces.

**38. (Currently Amended)** One or more tangible computer readable media having stored thereon a plurality of instructions that, when executed by a processor, cause the processor to:

create a first namespace with functions that enable identification of particular physical locations; and

create a second namespace with functions that are expected to be used by the first namespace and a plurality of additional namespaces, wherein the first namespace, the second namespace, and the plurality of additional namespaces are defined to organize a file system, the file system being included in a programming interface for developing programs.

**39. (Currently Amended)** One or more tangible computer readable media as recited in claim 38, wherein the instructions further cause the

processor to: create a third namespace with functions that enable documents to be described; create a fourth namespace with functions that enable identification of entities that can be contacted by a human being; and create a fifth namespace with functions common to multiple kinds of media.

**40. (Currently Amended)** One or more tangible computer readable media as recited in claim 39, wherein the instructions further cause the processor to: create a sixth namespace with functions specific to audio media; create a seventh namespace with functions specific to video media; and create an eighth namespace with functions specific to image media.

**41. (Currently Amended)** One or more tangible computer readable media as recited in claim 38, wherein the instructions further cause the processor to: create a third namespace with functions common to multiple kinds of media; create a fourth namespace with functions specific to audio media; create a fifth namespace with functions specific to video media; and create a sixth namespace with functions specific to image media.

**42. (Currently Amended)** A method comprising:

calling one or more first functions using a program development computer system that enable documents to be described; and

calling one or more second functions using the program development computer system that are core functions expected to be used by the one or more first functions as well as a plurality of additional functions, wherein the one or more first functions, the one or more second functions, and the plurality of additional functions are defined to organize a file system in the program development computer system, the file system being included in a programming interface.

**43. (Currently Amended)** A method as recited in claim 42, further comprising: calling one or more third functions common to multiple kinds of media using the program development computer system.

**44. (Currently Amended)** A method as recited in claim 43, further comprising: calling one or more fourth functions specific to audio media using the program development computer system; calling one or more fifth functions specific to video media using the program development computer system; and calling one or more sixth functions specific to image media using the program development computer system.

**45. (Currently Amended)** A method as recited in claim 42, further comprising: calling one or more third functions using the program development

computer system that enable identification of entities that can be contacted by a human being; and calling one or more fourth functions using the program development computer system that enable identification of particular physical locations.

**46. (Currently Amended)** A method as recited in claim 42, further comprising: calling one or more third functions specific to electronic mail messages using the program development computer system.

**47. (Currently Amended)** A method, comprising:  
receiving one or more calls to one or more first functions using a program development computer system that enable identification of entities that can be contacted by a human being; and

receiving one or more calls to one or more second functions that are core functions expected to be used by the one or more first functions as well as a plurality of additional functions using the program development computer system, wherein the one or more first functions, the one or more second functions, and the plurality of additional functions are defined to organize a file system in the program development computer system, the file system being included in a programming interface.



**48. (Currently Amended)** A method as recited in claim 47, further comprising: receiving one or more calls using the program development computer system to one or more third functions that enable documents to be described; receiving one or more calls using the program development computer system to one or more fourth functions common to multiple kinds of media; and receiving one or more calls using the program development computer system to one or more fifth functions that enable identification of particular physical locations.

**49. (Currently Amended)** A method as recited in claim 48, further comprising: receiving one or more calls using the program development computer system to one or more sixth functions specific to audio media; receiving one or more calls using the program development computer system to one or more seventh functions specific to video media; [[and]] receiving one or more calls using the program development computer system to one or more eighth functions specific to image media~~[[.]]~~ and receiving one or more calls using the program development computer system to one or more ninth functions specific to electronic mail messages.

**50. (Currently Amended)** One or more tangible computer readable media having stored thereon a plurality of instructions that, when executed by a processor, cause the processor to:

receive one or more calls to one or more first functions that enable identification of entities that can be contacted by a human being; and

receive one or more calls to one or more second functions common to multiple kinds of media, wherein the one or more first functions and the one or more second functions are defined to organize a file system, the file system being included in a programming interface for programming programs.

**51. (Currently Amended)** One or more tangible computer readable media as recited in claim 50, wherein the instructions further cause the processor to: receive one or more calls to one or more third functions that are core functions expected to be used by the one or more first functions, the one or more second functions, and a plurality of additional functions.

**52. (Currently Amended)** One or more tangible computer readable media as recited in claim 50, wherein the instructions further cause the processor to: receive one or more calls to one or more third functions that enable identification of particular physical locations; receive one or more calls to one or more fourth functions that enable documents to be described; and receive

one or more calls to one or more fifth functions specific to electronic mail messages.

**53. (Currently Amended)** One or more tangible computer readable media as recited in claim 50, wherein the instructions further cause the processor to: receive one or more calls to one or more third functions specific to audio media; receive one or more calls to one or more fourth functions specific to video media; and receive one or more calls to one or more fifth functions specific to image media.